ASTRA

A Space Charge Tracking Algorithm

Version 0.0   18-Sep-00

Table of Contents

1.0.0 Introduction

The Astra (**A S**pace Charge **Tr**acking **A**lgorithm) program package consists of the four parts:

1. The program *generator* which may be used to generate an initial particle distribution.
2. The program *Astra* which tracks the particles under the influence of external and internal fields.
3. The graphic program *fieldplot* which is used to display electromagnetic fields of beam line elements and space charge fields of particle distributions.
4. The graphic program *postpro* which is used to display phase space plots of particle distributions and allows a detailed analysis of the phase space distribution.
5. The graphic program *lineplot*, which is used to display the beam size, emittance, length etc versus the longitudinal beam line position or versus a scanned parameter, respectively.

The menu controlled graphic programs are based on the subroutine package PGPLOT[1]. They are basically self-explanatory, but some more details will be given in chapter 6.0.0.

The input files for the programs *generator* and *Astra* are organized in form of Fortran 90 namelists. Each namelist starts with an ampersand (&) followed by the name of the namelist and ends with a slash (/). An input file has to contain all namelists in a fixed order, but within the namelist only those parameters which are relevant have to be specified. Thus the minimal form of a namelist is:

```
&NAME
/
```

Within a namelist parameters are specified in the form: ‘name = Value’. The order of the parameters within a namelist is free. Specifications are separated by a comma or a line feed, with an arbitrary number of blanks or blank lines in between. Character input (keywords and file names) has to be enclosed by quotation marks (‘…‘). The input of keywords is not case sensitive. In general only the first character(s) are significant. Significant characters are indicated by bold letters in this manual.

[1] PGPLOT is a graphics subroutine library freely available for non-commercial use. For downloading and further information see: http//astro.caltech.edu/~tjp/pgplot .

2.0.0 Definition of the initial particle distribution

Rather than generating the initial particle distribution internally, the tracking programs *Astra* reads the initial particle coordinates from a file. This file may be generated by the program *generator* or by a user written program. However, also any output distribution of the *Astra* code can be used as input distribution, thus supporting the piecewise tracking of a long beam line. In order to be compatible with the graphic program *postpro* the input distribution file name should end with the extension ‘.ini’ or with ‘.zpos.run’, where zpos is a four digit number specifying the longitudinal beam position and run is a three digit number specifying the run number (see chapter 6.2.0). Table 1 lists the structure of particle distribution files. The Fortran format is: 8E12.4, 2I4.

Number	1	2	3	4	5	6	7	8	9	10
Parameter	x	y	z	px	py	pz	clock	macro charge	particle index	status flag
Unit	m	m	m	eV/c	eV/c	eV/c	ns	nC		

Table 1: Structure of particle distribution files.

The first line of the file defines the coordinates of the reference particle in absolute coordinates. It is recommended to refer it to the bunch center. All other particle coordinates are given relative to the reference particle! If the particles shall be emitted from a cathode they have to be generated with the same longitudinal position, e.g. z = 0.0 and with an appropriate spread in time, i.e. clock values in nanoseconds. In addition the status flag has to be set accordingly (see below).
The macro charge of the particle is given in nano Coulomb. It is possible to specify each particle with a different charge; the emittance calculation will be done with the appropriate weighting.
The particle index specifies the kind of particle to be tracked: Index 1 refers to electrons, 2 to positrons, 3 to protons and 4 to hydrogen ions. The sign of the charge is not relevant. It is possible to mix different kinds of particles as an initial particle distribution.
The status flag contains information of the particle status as listed in Table 2. Particles with a negative status flag are either lost by some mechanism or not yet started. (The output files lists the coordinates of all particles even of those, that have been lost.) Passive particles are tracked as normal particles but they are not taken into account in the calculation of the beam emittance etc and they are not taken into account when the space charge field is calculated. They will, however, be tracked taken the action of the space charge field onto them into account. They are typically used to cut off beam tails or halo particles. The trajectories of ‘probe particles’ and the space charge fields acting onto these particles will be found in an output file for later analysis.

Status flag	Comment	Status
-90	particle before Z_{min}, travelling backwards	lost
-20	passive probe particle , $r>r_{max}$	lost
-19	passive particle , $r>r_{max}$	lost
-17	trajectory probe particle, $r>r_{max}$	lost
-15	standard particle, $r>r_{max}$	lost
-6	passive probe particle, at the cathode	not yet started
-5	passive particle, at the cathode	not yet started
-3	trajectory probe particle at the cathode	not yet started
-1	standard particle, at the cathode	not yet started
0	passive probe particle	tracking but no output
1	passive particle	tracking but no output
3	trajectory probe particle	tracking
5	standard particle	tracking

Table 2: Definition of important status flags.

3.0.0 The program *generator*

The program *generator* generates an initial particle distribution file according to the previously described specifications.
The input file for *generator* has to have the extension ‘.in’. The default file name is ‘generator.in’. The input file consists of a single namelist named INPUT. A tabulated listing of all possible input parameters is given in chapter 7.0.0. The below listed input file: ‘generator1.in’ gives a simple example for the generation of a gaussian particle distribution:

```
&INPUT
  FNAME = 'Example.ini'
  Add=FALSE,  N_add=0,
  IPart=500,      Species='electrons'
  Probe=True,    Noise_reduc=T,          Cathode=F
  Q_total=1.0E0

  Ref_zpos=0.0E0,        Ekin=2.0E0

  Dist_z='gauss',           sig_z=1.0E0,    C_sig_z=2.0
  Dist_pz='g',              sig_pz=1.5,      cor_Ekin=0.0E0

  Dist_x='gauss',           sig_x=0.75E0,
  Dist_px='g',              Nemit_x=1.0E0,           cor_px=0.0E0
  Dist_y='g',               sig_y=0.75E0,
  Dist_py='g',              Nemit_y=1.0E0,           cor_py=0.0E0
 /
```

Running *generator* with this input file will result in the generation of the file ‘Example.ini’ containing the coordinates of 500 electrons with a total charge of 1 nC. Since ‘Cathode = F(alse)’ the particles are not emitted from a cathode and a longitudinal extension of the bunch has to be specified rather than a time spread. ‘Probe = True’ will result in the specification of six probe particles at the positions:

0.5 σx, 0.5 σz; 1.0 σx, 1.0 σz; 1.5 σx, 1.5 σz;

0.5 σy, -0.5 σz; 1.0 σy, -1.0 σz; 1.5 σy, -1.5 σz.

The trajectories of these particles and the space charge fields acting on these particles will be saved if ‘TrackS=True’ is set.

The specification ‘Noise_reduc = T(rue)’ forces the program to distribute the particles not randomly but quasi-randomly following a so-called Hammersley sequence. As a result statistical fluctuations are reduced, while at the same time artificial correlations are avoided which would be generated by a set up on a grid.

The longitudinal position of the bunch is at 0.0 m and the kinetic energy is 2.0 MeV. The longitudinal distribution is gaussian with an rms width of 1mm and a cut at 2 sigma. The distribution of the longitudinal momenta is uniform with an rms width of 1.5 keV. Alternatively it would be possible to specify a longitudinal emittance rather than the energy spread. No correlated energy spread is introduced.

The transverse distribution is gaussian in x and y with an rms width of 0.75 mm. The distribution of the transverse momenta is also gaussian and is set up in a way that the beam emittance will be 1 π mrad mm. No correlated beam divergence is introduced.

Besides the complete listing of all possible input parameters in chapter 7.0.0 a collection of the properties of different distributions can be found in Appendix??.

In order to assemble more complicated distributions it is possible to add up several distributions into a common file. In this case ‘Add=True’ and ‘N_add=n’ has to be specified, where n is the number of distributions to be added. Than the namelist INPUT has to be specified n times with different parameters. (FNAME, Add and N_add might be specified only once in the first namelist.) The reference particle of the combined distribution will be the reference particle defined in the first namelist.

If a dispersion is specified an energy correlated transverse offset will be added in a final step. The calculated emittance will hence be larger than specified in the input file.

After running *generator* the result can be visualized by calling *postpro* with the appropriate input argument, i.e. ‘postpro Example.ini’. See chapter 6.2.0.

Besides the file Example.ini the file NORRAN will be created by generator. It contains a new seed value for the random generator and will be updated every time generator is used.

4.0.0 The program *Astra*

The program Astra tracks particles through user defined external fields taking into account the space charge field of the particle cloud. The tracking is based on a Runge-Kutta integration of 4^{th} order with fixed time step.
The beam line elements are set up w.r.t. a global coordinate system in Astra. The axis of the (preferred) motion of the bunch is the z-axis (longitudinal axis). The horizontal plane is related to the x-axis, and the vertical plane is defined via the y-axis.
Astra is a three-dimensional code, but this first version allows the calculation of the space charge field only for round beams on a cylindrical grid. The longitudinal axis of this grid is set up parallel to the z-axis of the Astra coordinate system; therefore the orbit should have no large angle w.r.t. the z-axis when a space charge calculation is performed. For cylindrical elements (cavities, solenoids etc) also cylindrical coordinates are used, if convenient.
All calculations in *Astra* are done with double precision, while output and most input is in single precision.
The input file for *Astra* has to have the extension ‘.in’. The default file name is ‘rfgun.in’. The input of each class of beam line element is organized in a separated namelist. So far solenoids, cavities and quadrupoles can be treated. Besides the namelists for the beam line elements the namelist NEWRUN contains general instructions for the tracking, CHARGE contains the settings for the space charge calculation and SCAN contains instructions for the scanning routine. In the following an example input file without space charge will be discussed. In a second step the calculation of space charge fields will be included before finally the scanning routine will be described.

4.1.0 Example input file without space charge

```
&NEWRUN
  Head=’ Example 1 of ASTRA user’s manual’
  RUN=1
  Distribution = 'Example.ini’,     Xoff=0.0,   Yoff=0.0,
  TRACK_ALL=T,
  H_max=0.001,     H_min=0.0005
  Rmax=5.0E-3,

  ZSTART=0.0,     ZSTOP=1.5
  Zemit=30,       Zphase=1
  RefS=T
  EmitS=T,     PhaseS=T
  TrackS=F,    TcheckS=F
/

 &SCAN
 /

 &CHARGE
  LSPCH=F
  Nrad=10, Cell_var=2.0,      Nlong_in=10
  min_grid=0.5D-6
  Max_Scale=0.1
 /
```

```
 &CAVITY
  LEField=T,
  File_Efield(1)='efld.dat',
  Nue(1)=1.3,     MaxE(1)=50.0,     Phi(1)=0.0,
/

 &SOLENOID
  LBField=T,
  File_Bfield(1)='bfld.dat'
  MaxB(1)=0.2250
 /

 &QUADRUPOLE
 /
```

NEWRUN starts with a header string and a run number, which should be used to protocol different parameter settings. The run number will be found as an extension of all output files generated by *Astra*. The input particle distribution has been previously generated with *generator*. It is used without a transverse offset, i.e. on-axis. After the phasing of the cavities (see below) the reference particle, i.e. the first particle in the input distribution file, will be tracked through the beam line to check the beam line settings. In a second step the reference particle will be tracked again, starting with an offset at $x=x_{rms}$ and $y=y_{rms}$. If ‘TRACK_ALL=False’ is set the tracking will stop here. The maximum time step for the Runge-Kutta integrator is defined with the parameter H_max, while H_min is only active if the space charge fields are taken into account (see below). A particle will be discarded, if its radius (w.r.t. the z-axis of the coordinate system) exceeds Rmax.
The second set of parameters in NEWRUN is devoted to the generation of output. While the tracking starts at any position where the initial particle distribution is launched, output will be generated between ZStart and ZStop. The tracking will stop when the bunch position, i.e. the average position of all active particles is larger than ZStop.
The names of all files generated by *Astra* start with the project name, i.e. with the name of the input file and end with the run number. In between, separated by dots a type dependent name is given to the files. Table ?? in section?? gives a complete listing of all output files generated by *Astra* including the logical switches to start or suppress the generation of output.
Astra generates output on different length scales of the beam line. ‘RefS=True’ generates output of the off-axis reference trajectory, energy gain etc at each Runge-Kutta time step.
Output of the beam emittance and other statistical beam parameters is generated if ‘EmitS=True’. For the calculation of statistical bunch parameters the distance ZStop-ZStart is divided into Zemit intervals. Note, that the Runge-Kutta time step is adjusted, i.e. reduced if necessary, in order to interrupt the tracking close to the specified locations. (The beam position refers to the average longitudinal beam position.) This might lead to a reduction of each time step, i.e. to an increased accuracy of the calculation, if the intervals are shorter than the bunch motion in one time step. A warning is given in this case because the result of the calculation might depend on an output parameter if H_max is too big!
The complete particle distribution is saved at Zphase different locations if ‘PhaseS=True’. The distance ZStop-ZStart is divided into Zphase intervals and the nearest location defined by means of Zemit is chosen. Therefore it is recommended to

set Zemit=n·Zphase, $n \in N$. The approximate position is indicated in the file name as a four digit number, which corresponds in general to the rounded beam position in cm. If the distance between the intervals is below 1cm the number corresponds to the beam position in mm or even 10^{th} and 100^{th} of mm, respectively.
All namelists but NEWRUN start with a logical switch, which allows to deactivate all elements in that list, without changing other parameters. Hence, with ‘LSPCH=False’, the calculation of space charge forces is deactivated.
The namelist CAVITY allows to include rotational symmetric fields of standing wave cavities, as well as of travelling wave structures and electrostatic fields. The dependence of the longitudinal electric field component on the longitudinal position at the symmetry axis of the field has to be given in form of a table in case of standing wave structures and electrostatic fields. The radial and magnetic field components are deduced from the derivative of the longitudinal filed w.r.t. the longitudinal position. File_Efield(n) contains the name of the file where this table can be found, for the n^{th} cavity. Nue(n) states the frequency, MaxE(n) the **maximum** amplitude of the field and Phi(n) the phase of the wave. By default the energy gain of each cavity is scanned prior to the tracking of the reference particle in *Astra*. The user-defined phase refers than to the phase of the maximum energy gain in a cosine-like manner. Note, that the phase of the wave $\omega \cdot t + \varphi$ is increasing with time, i.e. the tail of the bunch ‘sees’ a higher phase than the head. Thus, in order to give the tail a higher energy than the head, one has to go to negative phases Phi(n). The phase, denoted by the program as a result of the auto-phasing procedure, is the number used internally. It refers to the phase of the wave at the time stamp of the reference particle when it is starting to be tracked. The auto-phasing can be switched off by setting ‘Auto_Phase=False’ in NEWRUN.
The namelist SOLENOID contains information about solenoid fields. Like in case of cavity fields a table of the longitudinal field component along the symmetry axis is required. Besides the file name of the table, only the **maximum** amplitude of the field has to be specified. (The scaling of the field to the defined amplitude might also be switched off by setting S_noscale=True.)
No Quadrupoles are used in this example.

4.2.0 Calculation of the space charge field

For the calculation of the space charge field a cylindrical grid (r, φ, z coordinates), consisting of rings in the radial direction and slices in the longitudinal direction, is set up over the extension of the bunch. The grid is Lorentz transformed into the average rest system of the bunch, where the motion of the particles is to good approximation non relativistic and a static field calculation can be performed by integrating numerically over the rings thereby assuming a constant charge density inside a ring. (The effect of the remaining particle motion in the average rest system will be discussed below.) The field contributions of the individual rings at the center points of the grid cells are added up and transformed back into the laboratory system. Here the field at any given point between the grid center points is calculated by means of a cubic spline interpolation, so that the field and the first derivatives w.r.t. the space coordinates are continuos functions. Outside of the grid a 1/r extrapolation is applied, so that the space charge field is defined (with reduced accuracy) over the whole space. For the tracking the space charge field is treated like the external field, i.e. a Runge-Kutta integration is performed based on the sum of all external and internal forces. It would, however, be too time consuming (and useless) to calculate the space

charge field on the grid center points again at every Runge-Kutta time step. Therefore an automatic procedure has been implemented which scales of the space charge field and the grid dimensions with the variation of the beam size, the beam energy etc. A new calculation of the field on the grid center points is initiated every time the scaling factor of the field exceeds a user-defined limit.

4.2.1. Set up of the grid

The space charge fields are calculated on a cylindrical grid, consisting of rings in the radial direction and of slices in the longitudinal direction. The grid is set up dynamically, i.e. the grid dimensions are based on the actual dimensions of the bunch. The user has to define the number of slices (parameter Nlong_in) and rings (parameter Nrad) that shall match exactly the dimensions of the bunch. For the calculation two more rings and four more slices are added outside of the bunch.
The total number of particles inside a ring scales, for a distribution with uniform charge density, linearly with the radius in the outside region of the bunch but more quadratically at the innermost region. As an example consider the case of 5000 particles uniformly distributed inside a radius of 1.5mm and a grid of 10 rings. In the innermost ring only about 50 particles will be located, while in the outermost ring about 950 particles will be counted. When these particles are additionally distributed into 10 longitudinal slices, half of the innermost rings will have no particle at all! Furthermore one has to take into account, that the field of a charge ring acts predominantly to the outside of the ring. Thus the field in the outside region of the bunch is composed by more or less all particles while this is not the case in the central region of the bunch. To counteract this unfavorable situation it is possible to vary the radial grid height over the bunch radius by means of the parameter Cell_var in Astra. If for example ‘Cell_var=2’ is chosen, the innermost ring will be twice as high as the outermost ring. In order to get a sufficient statistical accuracy it is nevertheless necessary to choose a small number of grid cells. The description of the fields, however, will have a better resolution than suggested by the grid size due to the smooth interpolation algorithm.
The space charge fields generated by a particle distribution can be visualized with the program *fieldplot*. (See chapter 6.3.0.) Here also the number of grid cells and the cell variation may be changed and optimized.

4.2.2. The emission of particles from a cathode

In order to simulate the emission of particles from a (plane) cathode the particles are started from the cathode according to the timing spread of the initial particle distribution. While the step size of each particle is adjusted accordingly, the space charge field is updated with the user-defined step size H_min (in the input deck). If during the emission the space charge field shall be updated n times, H_min has to be T/n where T is the total emission time. In general H_min is an uncritical parameter if n is reasonably high. It is however useless to set n very high if the number of particles is not very high.
Since the number of particles is small during the first steps of the emission process the number of longitudinal slices shall be reduced during the emission by means of the parameter min_grid, which defines the minimal grid size in longitudinal direction. Whenever the bunch length is so small, that the grid size would be below min_grid, the number of slices is adjusted accordingly. The bunch length s_{min} after the first time step, t=H_min has been performed can be estimated as:

$$s_{\min} = \frac{e \cdot E}{2m} \cdot t^2$$

where E is the accelerating field strength and m is the rest mass of the particles in the bunch.
If min_grid is set to this value the number of slices increases smoothly as the bunch length and the number of particles during the emission process.
For the set up of the grid also particles which are not emitted yet are taken into account, in order to avoid too small radii during the first steps of the process.
By default the mirror charge of the bunch at the cathode plane is taken into account. The fields of the ‘mirror bunch’ are calculated in the rest system of the mirror bunch at the Lorentz transformed distance between bunch and mirror bunch, transformed back into the laboratory system and added to the field of the bunch. The calculation of the mirror charge is switched off when the contribution of the mirror bunch field at two positions of the bunch (in the center and in the tail of the bunch) is below 1‰ of the bunch field. The calculation of the mirror charge can be suppressed by setting ‘Lmirror=FALSE’. Retarded time effects and radiation effects are not taken into account in this version.

4.2.3. The effect of particle motions in the average rest system

For the calculation of the space charge field the particle coordinates and grid dimensions are Lorentz transformed into the average rest system of the bunch. In an rf gun a significant longitudinal velocity spread is build up at the cathode. Transverse motions are boosted (approximately with γ) when transformed into the average rest system, so that the particles are in general not at rest in the average rest system. Astra calculates the correlated motion β_r, β_φ, and β_z in the average rest system of the bunch for each grid cell (β_φ is build up in presence of solenoid fields). Since γ is still close to one in the average rest system (β<0.3 has been found in an L-Band rf gun) the calculation of the electric field is not affected, i.e. the field is not deformed due to relativistic velocities. The particle motions induce, however, additional magnetic field components. The calculation of these components by means of Biot-Savarts rule leads basically to the same integrals that have to be solved for the calculation of the electric field components. Therefore the calculation of magnetic fields in the average rest system is implemented in Astra. The magnetic components have an effect on the standard field components Er, Bφ and Ez and generate the additional components Br, Eφ and Bz. However, no effect on the beam emittance or any other beam parameter could be observed so far due to the cancellation of Er with Bφ and of Eφ with Br. By default the calculation of the magnetic field components in the average rest system is switched off. It can be switched on by setting ‘Linert=True’.

4.2.4. Scaling of the space charge field

The space charge field of the bunch is (at least at sufficient high energies) a slowly with time varying function. Hence, instead of calculating new field coefficients at every time step, it is justified to scale the field coefficients according to:

$$Er \propto \frac{Q}{Q_0} \cdot \left(\frac{\sigma r_0}{\sigma r}\right)^{nr(r)} \cdot \left(\frac{\sigma z_0}{\sigma z}\right)^{nr(z)} \cdot \left(\frac{\gamma}{\gamma_0}\right)^{nr(\gamma)}$$

$$Ez \propto \frac{Q}{Q_0} \cdot \left(\frac{\sigma r_0}{\sigma r} \right)^{nz(r)} \cdot \left(\frac{\sigma z_0 \cdot \gamma_0}{\sigma z \cdot \gamma} \right)^{nz(z\cdot\gamma)}$$

$$B\varphi \propto Er \cdot \frac{\beta}{\beta_0}$$

where Q/Q_0, $\sigma r/\sigma r_0$, $\sigma z/\sigma z_0$, γ/γ_0 and β/β_0 denote the relative variation of the bunch charge, the bunch radius, length, energy and velocity, respectively.

At the same time the grid has to be scaled with the variation of the radial bunch size and the bunch length.

$nr(r)$, $nr(z)$, $nr(\gamma)$, $nz(r)$ and $nz(z \cdot \gamma)$ are functions that depend on the aspect ratio $A = {}^{\sigma z \cdot \gamma}\!/_{\sigma r}$ of the bunch in the rest system. They are constant for *A >> 1*.

In case of a pancake like bunch the fields are proportional to:

$$Er \propto \frac{Q}{Q_0} \cdot \left(\frac{\sigma r_0}{\sigma r} \right)^2 \cdot \frac{\gamma}{\gamma_0}$$

$$Ez \propto \frac{Q}{Q_0} \cdot \left(\frac{\sigma r_0}{\sigma r} \right)^2 \cdot \frac{\sigma z_0 \cdot \gamma_0}{\sigma z \cdot \gamma}$$

while a cigar like bunch scales like:

$$Er \propto \frac{Q}{Q_0} \cdot \frac{\sigma r_0}{\sigma r} \cdot \frac{\sigma z_0}{\sigma z}$$

$$Ez \propto \frac{Q}{Q_0} \cdot \left(\frac{\sigma z_0 \cdot \gamma_0}{\sigma z \cdot \gamma} \right)^2$$

The functions used in *Astra* are only approximate functions that fit roughly numerical results for a number of bunch dimensions with uniform longitudinal and radial distribution. However, the scaling is in any case better than assuming a constant field. If the scaling factor for the radial or longitudinal electric field exceeds a user-defined limit a new space charge calculation is initiated. The parameter Max_scale defines this limit; if for example ‘Max_scale=0.05’ the scaling factor has to be between 1.05 and 0.95, respectively.
Strong changes of the particle distribution, without variation of the bunch dimensions, cannot be taken into account by the scaling routine and have to be treated separately. This effect is observed during the compensation process of space charge induced emittance growth close to the emittance minimum, when the particles in the bunch center move outwards, while the particles in the head and the tail of the bunch are still moving inward. In order to limit the number of scalings in this case, the parameter Max_count can be set.
As long as particles are emitted from a cathode the scaling procedure is not active. Instead the space charge fields are updated after each time step H_min.
The effectiveness of the scaling and the setting of Max_scale can be controlled by setting ‘TrackS=True’. A file will be logged that contains the trajectories of particles marked as probe particles and the space charge fields acting onto these particles. Both

can be displayed with the program *lineplot*. (While *Astra* can deal with any number of probe particles, *lineplot* will support the display of only up to ten particles.) Since the data are logged at each Runge-Kutta time step the generated files tend to be large. Therefore one might consider using this option only for short critical sections and not for long beam lines. Additional information about the scaling procedure can be gained if the option ‘TcheckS=True’ is set. A file will be logged that contains the scaling factors at each time step separated into the contributions $\left(\frac{\sigma r_0}{\sigma r}\right)^{nr(r)}$, $\left(\frac{\sigma z_0}{\sigma z}\right)^{nr(z)}$, $\left(\frac{\gamma}{\gamma_0}\right)^{nr(\gamma)}$ etc, which allows to identify the driving force of the space charge field development. Also the scaling counter, which counts how often the field is scaled before it is updated and the time between updates of the space charge fields is saved. The later one is used to calculate the average Runge-Kutta time step. See the description of *lineplot* chapter 6.1.2 for displaying results.

4.3.0 Scanning and optimizing beam parameters

Astra offers different options for doing parameter scans. A simple, predefined scan based on a single particle tracking (the reference particle) is performed by setting ‘PHASE_SCAN=True’ in NEWRUN. The energy gain as function of the cavity phase is stored as well as the bunch compression factor, i.e. the ratio of the bunch length at the exit of the cavity to the bunch length at the entrance of the cavity. From the derivative of the energy gain w.r.t. the cavity phase a quantity is derived which is proportional to the rf induced energy spread. (See chapter 6.1.2.) When the scan for one cavity is finished, the reference particle will be tracked through the cavity on the user-defined phase up to the entrance of the next cavity. Thus, for low energy beams ($\beta < 1$), the result of the scan depends on the user-defined phase.
User defined scans can be performed with the scanning procedure defined in the namelist SCAN. The following example shows a setting for a scan of the cavity gradient of cavity number one:

```
&SCAN
  LScan=T
  Scan_para='MaxE(1)'
  S_min=10.0,     S_max=50.0, S_numb=5
  FOM(1)='hor. Emittance'
  FOM(2)='long. Emittance'
  FOM(3)='bunch length'
 /
```

(All valid scanning parameters are written in italic letters in the namelist tables of chapter 8.0.0.) Besides the minimum and maximum set point of the scanning parameter, S_min and S_max, the total number of set points S_numb has to be specified. The user also has to define which beam parameters shall be stored. Up to 10 different output parameters can be specified (FOM(1) to FOM(10)). See chapter 8.2.0 for a listing of valid keywords. The standard output generation (emittance vs. z etc) is suppressed when a scan is performed.
While with the previously discussed set up all parameters, as defined with FOM(1) to FOM(10), are saved at the end of the beam line, it is also possible to search for a minimum or maximum within a predefined longitudinal range with the scanning routine. In this case ‘L_min=True’ or ‘L_max=True’ has to be specified. The minimum or maximum value of FOM(1) within the longitudinal interval S_zmin to S_zmax will be saved. The interval is divided into S_dz subintervals. At the end of

each subinterval the emittance etc is calculated and the optimum is updated. The values of FOM(2) to FOM(10) will be saved at the position of the optimum of FOM(1). The position of the optimum will also be saved.
Another possibility for parameter scans is given with the looping option. All namelists start with the parameters LOOP and NLoop. If ‘LOOP=True’ is set, NLoop has to be set to a positive integer value. The complete namelist for which the looping has been activated has to be specified NLoop times with varying parameters. *Astra* will perform NLoop complete tracking calculations, working successively through the specified namelists and incrementing the run number automatically. It is possible to specify ‘NLOOP=True’ simultaneously in more than on namelist, but only with the same value for NLoop.

5.0.0 Organization of output files

As previously described *Astra* produces output on different length scales, time scales or scales for the variation of a parameter, respectively. Table 3 lists generic file names, logical switches and the scale on which data are stored. Table 4 lists the output file data structure, i.e. the parameters that can be found in the different files, their units and the format of the files.

generic name	logical switch	approx. scale
project.ref.run	RefS	Runge-Kutta time step $H = H_{max}$
project.track.run	TrackS	Runge-Kutta time step H
project.cathode.run	CathodeS	Runge-Kutta time step H
project.tcheck.run	TcheckS	time between updates of the space charge field
project.Xemit.run project.Yemit.run project.Zemt.run	EmitS	(ZStop-ZStart)/Zemit
project.Cemit.run	C_EmitS	(ZStop-ZStart)/Zemit
project.zpos.run	PhaseS	(ZStop-ZStart)/Zemit
project.Log.run	PhaseS	(ZStop-ZStart)/Zemit
project.LandF.run	PhaseS	(ZStop-ZStart)/Zemit
project.PScan.run	Phase_scan	1 degree of the rf phase of each cavity
project.Scan.run	LScan	(S_max-S_min)/S_numb
project.lab.run	LScan	

run = run number, zpos = z-position

Table 3: Generic file names, logical switches and scales for the generation of output with *Astra*.

Name	1	2	3	4	5	6	7	8	9	Format
ref	z m	t ns	pz MeV/c	dE/dz Mev/m	x_{off} mm	y_{off} mm				6E12.4
track	part. numb	stat. flag	z m	x mm	y mm	Ez V/m	Er V/m	Bφ V/m		2I5,6E12.4
Cathode	z m	t ns	Ez on cathode V7m							
tcheck	z m	t ns	$\frac{\sigma_{r0}}{\sigma_r}^{nr(r)}$	$\frac{\sigma_{z0}}{\sigma_z}^{nr(z)}$	$\frac{\gamma}{\gamma_0}^{nr(\gamma)}$	$\frac{\sigma_{r0}}{\sigma_r}^{nz(r)}$	$\frac{\sigma_{z0}\cdot\gamma_0}{\sigma_z\cdot\gamma}^{nz(\gamma z)}$	scaling counter		7E12.4,I10
Xemit	z m	t ns	x_{avr} mm	x_{rms} mm	x'_{rms} mrad	$\varepsilon_{x,norm}$ π mrad mm	$x\cdot x'_{avr}$ mrad			7E12.4
Yemit	z m	t ns	y_{avr} mm	y_{rms} mm	y'_{rms} mrad	$\varepsilon_{y,norm}$ π mrad mm	$y\cdot y'_{avr}$ mrad			7E12.4
Zemit	z m	t ns	E_{kin} Mev	z_{rms} mm	ΔE_{rms} kev	$\varepsilon_{z,norm}$ π keV mm	$z\cdot E'_{avr}$ keV			7E12.4
Cemit	z m	$\varepsilon_{x,norm\cdot}$, Cx_95, Cx_90, Cx_80 π mrad mm			$\varepsilon_{x,norm\cdot}$, Cy_95, Cy_90, Cy_80 π mrad mm			$\varepsilon_{x,norm\cdot}$, Cz_95, Cz_90, Cz_80 π keV mm		13E12.4
zpos*2										8E12.4,2I4
Log	file name	z m	Bz T							A30,2D12.4
LandF	z m	Npart*1	Q nC	Lost(1)	Lost(2) J	Lost(3) J				6E12.4
PScan	phase deg	E_{kin} MeV	compression factor	β/β_0						4E12.4
Scan	Scan_ para	z m	FOM(1) – FOM(10)							12E12.4
lab	Label for plots: X-axis, Y-axis, title									A80

*1 Npart = number of active particles   *2 zpos files have the same structure as input distribution files

Table 4 Output file data structure.

6.0.0 Graphics programs

6.1.0 The program *lineplot*

The program *lineplot* allows to display a number of beam parameters as beam emittance, bunch length etc versus the longitudinal beam position or versus a scanned parameter. Start the program with ‘lineplot [project name],[run number]’. The project name is the name of the *Astra* input file. If no project name is specified the default name: rfgun is assumed. The run number has to be separated from the project name by a comma without blanks and has to have three digits. If no run number is specified run number 001 is assumed. When starting the code two windows, one for displaying and one for selecting from the menu will come up. The menu contains only ‘dishes’ of which parameter files exist, i.e. if ‘EmitS=False’ has been chosen in the *Astra* input file no emittance plot will be offered. By clicking with the left mouse button onto the circles different parameters can be displayed. By clicking onto the word ‘menu’ different menus will be shown. Note, that additional information will be displayed in the window from which *lineplot* was started. Here also input might be requested.

6.1.1. Menu 1

1 Trans. Emittance: Displays the horizontal (black) and vertical (red) rms emittance of the beam along the z-axis. The calculation of the emittance is based on the statistical formula of Lapostolle. Canonical transverse momenta are used, except if Lmagnetized =True is set.

2 rms Beam Size: Displays the horizontal (black) and vertical (red) rms beam size of the beam along the z-axis.

3 rms Divergence: Displays the horizontal (black) and vertical (red) rms divergence of the beam along the z-axis.

4 Long. Emittance: Displays the longitudinal rms emittance of the beam along the z-axis. The calculation is based on a statistical approach equivalent to the transverse case.

5 rms Bunch Length: Displays rms bunch length of the beam along the z-axis.

6 rms Energy Spread: Displays the rms energy spread of the beam along the z-axis.

7 cor. Energy Spread: Displays the correlated part of the energy spread of the beam along the z-axis.

8 average Energy: Displays the average energy of the beam along the z-axis.

9 Particle velocity: Displays the average velocity of the beam along the z-axis.

10 Reference particle momentum: Displays the momentum of the reference particle along the z-axis. The data are based on the first, single particle tracking in *Astra*.

11 dp/dz: Displays the momentum gain of the reference particle along the z-axis, i.e. it reproduces the accelerating field as seen by the particle. The data are based on the first, single particle tracking in *Astra*.

12 Ref. particle trajectory: Displays the off-axis trajectory of the reference particle along the z-axis. The data are based on the first, single particle tracking in *Astra*, i.e. without space charge.

13 Trajectories x vs. z; y vs. z: Displays the trajectories of the probe particles along the z-axis in cartesian coordinates. At maximum 10 trajectories can be displayed.

14 Trajectories r vs. z; x vs. y: Displays the trajectories of the probe particles along the z-axis in cylindrical coordinates. At maximum 10 trajectories can be displayed.

15 Space charge fields: Displays the space charge fields (Er and Ez) acting onto the probe particles along the z-axis. At maximum the fields of 10 particles can be displayed.

18 ZOOM: Allows to zoom into all plots.

19 fit menu: The sub menu fit allows to fit a number of analytical functions to the presently displayed data. The data may also be saved to a reference file or red from a reference file for comparison, respectively. The fit menu is not accessible from all plots.

20 to ps file: Generates a post script file of the present plot for printing etc.

21 overview to ps file: Generates a post script file which contains the first 9 plots of the menu page on one page. Uses the default name: (project name).ps.

22 next run: Increases the run number by one.

23 previous run: Decreases the run number by one.

24 Exit: Exits *lineplot*.

6.1.2. Menu 2

1 Energy vs. Phase: Displays the energy gain versus the rf phase for all cavities in one plot. See chapter 4.3.0.

2 dE/dz vs. Phase: Displays the derivative of the energy gain versus the rf phase, which is proportional to the correlated energy spread, for all cavities. See chapter 4.3.0.

3 Compression factor (z): Displays the ratio of the final bunch length, i.e. at the exit of the cavity to the initial bunch length, i.e. at the entrance of the cavity for all cavities. In case of a gun cavity, the initial bunch length is replaced by the emission time · the velocity of light. See chapter 4.3.0.

4 Compression factor (t): Displays the ratio of the final bunch length to the initial bunch length times the inverse ratio of the particle velocities for all cavities. In case of a gun cavity, the initial bunch length is replaced by the emission time. See chapter 4.3.0.

5 Particle loss: Displays the number of lost particles per meter along the z-axis. The intervals are determined by the parameter Zphase.

6 Energy deposition: Displays the energy deposited in the surrounding structure by lost particles per meter along the z-axis. The intervals are determined by the parameter Zphase.

7 Beam loading: Displays the total energy exchange of the bunch with external fields per meter along the z-axis. The intervals are determined by the parameter Zphase.

9 Sp. ch. scaling factors: Displays the space charge scaling factors as described in chapter 4.2.4.

10 Sp. ch. scaling counter: Displays the space charge scaling factors as described in chapter 4.2.4.

11 Average time step: Displays the average Runge-Kutta time step between updates of the space charge fields.

19 ZOOM: See chapter 6.1.1.

20 fit menu: See chapter 6.1.1.

21 to ps file: See chapter 6.1.1.

22-24: See chapter 6.1.1.

6.1.3. Menu 3

1 to 10 FOM(1) to FOM(10): Displays the result of a scanning procedure according to the *Astra* input deck. See chapter 4.3.0.

11 position: Displays the longitudinal position at which the values of FOM(1) to FOM(1) were saved.

17 change title: Allows to change the labels and the title of the displayed plot.

18-24: See chapter 6.1.1.

6.2.0 The program *pospro*

The program *postpro* allows to display different phase space plots and to perform a detailed phase space analysis. *Postpro* requires as input arguments the project name, the run number and the z-position to be displayed at the start up. Within *postpro* one can step through the different z-positions at which the complete phase space distribution has been saved and through the different run numbers. Missing input arguments are completed by default values:

project name: rfgun
run number: 001
z-position: last saved distribution in the run

Thus valid calls are: ‘*postpro*’, ‘*postpro name’*, ‘*postpro name,run*’, ‘*postpro.zpos’*, ‘*postpro name.zpos,run’* or *‘postpro name.zpos.run’*. (Here name is the project name, zpos represents a four digit number for the z-position and run represents a three digit number.) Finally it is possible to display any distribution with the extension ‘*.ini’*, e.g. input files generated by *generator*.
When starting the code two windows, one for displaying and one for selecting from the menu will come up, even if no data exists for the specified arguments. By clicking with the left mouse button onto the circles different phase space plots can be displayed. Note, that additional information will be displayed in the window from which *postpro* was started. Here also input might be requested. For the plots a color and symbol code is used referring to the particle status as shown in Table 5. However, all plots except z-plot will not display particles which have been lost.

Particle status	**Status flag**	**color & symbol**
normal particle	>1	black stars
passive particle	0, 1	green crosses
particle at the cathode	-1 to -6	brown squares
particle lost with $r > r_{max}$	-12 to –20	red triangles
particle lost, travelling backwards	-20	blue circles

Table 5: Color and symbol code for *postpro* plots

6.2.1. Menu

1 Trans. phase space: Displays the horizontal and vertical phase space and the projection onto the horizontal and vertical axis.

2 Long. phase space (z): Displays the longitudinal phase space, the projection onto the longitudinal axis and the momentum distribution.

3 Long. phase space (t): The same as plot 2 but with the time as coordinate instead of the z-position.

4 Front, Top & Side view: Displays a view of the bunch from different directions.

5 Front, Top & Side view vs. time: The same as plot 4 but with the time as coordinate instead of the z-position.

6 core emittance: Displays the transverse and longitudinal rms emittance as function of the number of particles taken into account.

7 slice emittance: Displays the phase space for ten slices of the bunch.

10 z-plot: Displays the position of all particles along the beam line.

11 fit menu: See chapter 6.1.1.

12 forward: Moves to the next saved beam position down stream. At the end of the beam line *postpro* circles back to the first saved distribution.

13 backward: Moves to the next saved beam position up stream. At the beginning of the beam line *postpro* circles back to the last saved distribution.

15 to ps file: See chapter 6.1.1.

16 next run: See chapter 6.1.1.

17 previous run: See chapter 6.1.1

18 Exit: Exits *postpro*.

6.3.0 The program *fieldplot*

With the program *fieldplot* the electric and magnetic fields of beam line elements as well as the space charge field of saved particle distributions can be displayed. (Particle distributions to be emitted from a cathode have no longitudinal extension, hence no space charge field can be displayed.) *Fieldpot* refers to the *Astra* input file, i.e. only the project name has to be specified as input argument.
When starting the code two windows, one for displaying and one for selecting from the menu will come up. By clicking with the left mouse button onto the circles different plots can be displayed. Note, that additional information will be displayed in the window from which *fieldplot* was started. Here also input might be requested.

6.3.1. Menu

1 Cavity fields: Displays the amplitude of the longitudinal electric field of all cavities in the beam line. By clicking onto ‘next page’ the radial electric field amplitude and the azimuthal magnetic field amplitude can be displayed.

2 Solenoid fields: Displays the longitudinal magnetic field and the radial field gradient of all solenoids in the beam line in one plot. By clicking onto ‘next page’ both components can be displayed alone.

3 Quadrupole fields: Displays the horizontal and vertical field gradient of all quadrupoles in the beam line. By clicking onto ‘next page’ the longitudinal magnetic field and a combined plot of all components can be displayed, respectively.

10 z-plot: allows to display the space charge field of saved particle distributions. A sub menu will come up, see chapter 6.3.2.

12 fit menu: See chapter 6.1.1.

13 ZOOM: See chapter 6.1.1.

14 next page: switches between different field components to be displayed.

15 to ps file: See chapter 6.1.1.

16 Exit: Exits *fieldplot*.

6.3.2. Sub Menu for Space Charge Fields

1 Er: Displays the radial electric field component of the space charge field.

2 Efi: Displays the azimuthal electric field component of the space charge field. See chapter 4.2.3.

3 Ez: Displays the longitudinal electric field component of the space charge field.

4 Br: Displays the radial magnetic field component times the velocity of light of the space charge field. See chapter 4.2.3.

5 Bfi: Displays the azimuthal magnetic field component times the velocity of light of the space charge field.

6 Bz: Displays the longitudinal magnetic field component times the velocity of light of the space charge field. See chapter 4.2.3.

7 Er eff: Displays Er - $\beta c \cdot$Bfi

8 change number of lines: requires input to change the number of lines to be displayed.

9 radial: field components are shown as function of the radial position.

10 longitudinal: field components are shown as function of the longitudinal position.

11 change grid: requires input to change the number of radial and longitudinal grid cells and of the cell variation. See chapter 4.2.1.

12 scale energy: allows to scale the energy for a given distribution.

13 change distribution: requires input to change the particle distribution to be displayed.

14 mirror charge on/off: switches the mirror charge on or off.

17 to ps file: See chapter 6.1.1.

18 Exit sub menu: Returns to *fieldplot*.

7.0.0 Input namelist for *generator*

The keyword ‘radial’ specifies a distribution with uniform charge density within a circle of 2.0 times sig_x or 2.0 times sig_px, respectively.
The keyword ‘plateau distribution’ specifies a distribution with a soft rising at both ends and a plateau in between. Besides the length of the distribution (FWHM) the rising can be specified.

Parameter	Specification	Unit	Default Value
FNAME	Character*80		rfgun.ini
	file name for initial particle distribution; recommended extension are .ini or .zpos.run, respectively.		
Add	Logical		FLASE
	if true the input list has to be specified N-add times and N_add different distributions will be added.		
N_add	Integer		0
	number of distributions to be added.		
Ipart	Integer		100
	number of particles to be generated.		
Species	Character*80		Electrons
	species of particles to be generated. Valid are: **El**ectrons, **po**sitrons, **pr**otons and **hy**drogen ions.		
Probe	Logical		TRUE
	if true, 6 probe particles are generated at locations: 0.5σx, 0.5σz; 1.0σx, 1.0σz; 1.5σx, 1.5σz; 0.5σy, -0.5σz; 1.0σy, -1.0σz; 1.5σy, -1.5σz.		
Passive	Logical		FALSE
	if true only passive particles will be generated.		
Noise_reduc	Logical		TRUE
	if true, particle coordinates are generated quasi-randomly following a Hammersley sequence.		
Cathode	Logical		TRUE
	if true the particles will be generated with a time spread rather than with a spread in the longitudinal position, i.e. sig_z is set to zero and sig_clock has to be specified. Status flags will be set accordingly.		
Q_total	Real*4	nC	1.0
	total charge of the particles. The total charge is equally distributed on the Npart particles. (*Astra* would allow also particles with different macro charges in one distribution.)		
Ref_zpos	Real*4	m	0.0
	z position of the reference particle, i.e the longitudinal bunch position.		
Ref_clock	Real*4	ns	0.0
	initial clock value of the reference particle, can in general be set to zero.		

Ref_Ekin	Real*4	MeV	0.0
	initial kinetic energy of the reference particle. If zero a minimum kinetic energy of 10^{-18}eV is specified for numerical reasons.		
Dist_z	Character*80		uniform
	specifies the longitudinal particle distribution. Valid are **g**aussian, **u**niform and **p**lateau distribution.		
sig_z	Real*4	mm	0.0
	rms value of the bunch length.		
C_sig_z	Real*4		100.0
	cuts off the longitudinal distribution at C_sig_z times sig_z.		
Lz	Real*4	mm	0.0
	length of the bunch; only for plateau distribution.		
rz	Real*4	mm	0.0
	rising of the bunch distribution; only for plateau distribution.		
sig_clock	Real*4	ns	1.0D-3
	rms value of the emission time, i.e. the bunch length if generated from a cathode.		
C_sig_clock	Real*4		100.0
	cuts off the longitudinal distribution at C_sig_clock times sig_clock.		
Lt	Real*4	ns	0.0
	length of the bunch; only for plateau distribution.		
rt	Real*4	ns	0.0
	rise time of the bunch; only for plateau distribution.		
Dist_pz	Character*80		uniform
	specifies the longitudinal energy and momentum distribution, respectively. Valid are **g**aussian, **u**niform and **p**lateau distribution.		
sig_Ekin	Real*4	keV	0.0
	rms value of the energy spread.		
C_sig_Ekin	Real*4		100.0
	cuts off the energy and momentum distribution at C_sig_Ekin times sig_Ekin.		
LE	Real*4	keV	0.0
	width of the energy distribution; only for plateau distribution.		
rE	Real*4	keV	0.0
	rising of the energy distribution; only for plateau distribution.		
emit_z	Real*4	π keV mm	0.0
	longitudinal particle emittance. Can be specified instead of the energy spread. If an energy spread and an emittance is specified the energy spread has priority.		
cor_Ekin	Real*4	keV	0.0
	correlated energy spread.		

Dist_x	Character*80		gaussian
	specifies the transverse particle distribution in the horizontal direction. Valid are **g**aussian, **u**niform, **r**adial and **p**lateau distribution.		
sig_x	Real*4	mm	1.0
	rms bunch size in the horizontal direction. Also the vertical bunch size if Dist_x = radial.		
C_sig_x	Real*4		100.0
	cuts off the horizontal distribution at C_sig_x times sig_x. Cuts off also the vertical distribution if Dist_x = radial.		
Lx	Real*4	mm	0.0
	width of the horizontal particle distribution; only for plateau distribution.		
rx	Real*4	mm	0.0
	rising of the horizontal particle distribution; only for plateau distribution.		
Disp_x	Real*4	m	0.0
	horizontal dispersion; a horizontal offset is added to all particles according to: $x = x + Disp\_x \cdot \frac{\Delta P}{P}$; increases the calculated bunch emittance.		
Dist_px	Character*80		gaussian
	specifies the transverse momentum distribution in the horizontal direction. Valid are **g**aussian, **u**niform, **r**adial and **p**lateau distribution.		
Nemit_x	Real*4	π mrad mm	0.0
	normalized transverse emittance in the horizontal direction. Also the normalized vertical emittance if Dist_px = radial.		
C_sig_px	Real*4		100.0
	cuts off the horizontal momentum distribution at C_sig_px times sig_px. Cuts off also the vertical momentum distribution if Dist_px = radial.		
Lpx	Real*4	eV/c	0.0
	width of the horizontal momentum distribution; only for plateau distribution.		
rpx	Real*4	eV/c	0.0
	rising of the horizontal momentum distribution; only for plateau distribution.		
cor_px	Real*4	mrad	0.0
	correlated beam divergence in the horizontal direction $= -\frac{\alpha}{\beta[m]} \cdot x_{rms}[mm]$.		
Dist_y	Character*80		gaussian
	specifies the transverse particle distribution in the vertical direction. Valid are **g**aussian, **u**niform and **p**lateau distribution; not significant if Dist_x = radial.		
sig_y	Real*4	mm	1.0
	rms bunch size in the vertical direction. Not significant if Dist_x = radial.		
C_sig_y	Real*4		100.0
	cuts off the vertical distribution at C_sig_y times sig_py. Not significant if Dist_x = radial.		

Ly	Real*4	mm	0.0
	width of the vertical particle distribution; only for plateau distribution.		
ry	Real*4	mm	0.0
	rising of the vertical particle distribution; only for plateau distribution.		
Disp_y	Real*4	m	0.0
	vertical dispersion; a vertical offset is added to all particles according to: $y = y + Disp\_y \cdot \frac{\Delta P}{P}$; increases the calculated bunch emittance.		
Dist_py	Character*80		gaussian
	specifies the transverse momentum distribution in the vertical direction. Valid are **g**aussian, **u**niform and **p**lateau distribution; not significant if Dist_px = radial.		
Nemit_y	Real*4	π mrad mm	0.0
	normalized transverse emittance in the vertical direction. Not significant if Dist_px = radial.		
C_sig_py	Real*4		100.0
	cuts off the vertical momentum distribution at C_sig_py times sig_py. Not significant if Dist_px = radial.		
Lpy	Real*4	eV/c	0.0
	width of the vertical momentum distribution; only for plateau distribution.		
rpy	Real*4	eV/c	0.0
	rising of the vertical momentum distribution; only for plateau distribution.		
cor_py	Real*4	mrad	0.0
	correlated beam divergence in the vertical direction $= -\frac{\alpha}{\beta[m]} \cdot y_{rms}[mm]$.		

8.0.0 Input namelists for *Astra*

8.1.0 The namelist NEWRUN

The namelist NEWRUN contains basic instructions for the tracking, as well as specifications for the output generation.

Parameter	Specification	Unit	Default Value
LOOP	Logical		FALSE
	see chapter 4.3.0.		
NLoop	Integer		
	see chapter 4.3.0.		
Head	Character*80		
	header line for protocol.		
RUN	Integer		1
	the RUN number is used as extension for all output files (see Table?). It is automatically incremented if a LOOP is specified.		
Distribution	Character*80		
	name of the initial particle distribution, see chapters 2.0.0 and 3.0.0.		
Xoff	Real*8	mm	-1.0
	horizontal offset of the input distribution. Active if Xoff $\geq$ 0.0.		
Yoff	Real*8	mm	-1.0
	vertical offset of the input distribution. . Active if Yoff $\geq$ 0.0.		
Xrms	Real*8	mm	-1.0
	horizontal rms beam size. Scaling is active if Xrms > 0.0.		
Yrms	Real*8	mm	-1.0
	vertical rms beam size. Scaling is active if Xrms > 0.0.		
XYrms	Real*8	mm	-1.0
	horizontal and vertical rms beam size. Scaling is active if XYrms > 0.0.		
Zrms	Real*8	mm	-1.0
	rms bunch length. Scaling is active if Zrms > 0.0.		
Trms	Real*8	ns	-1.0
	emission time of the bunch. Scaling is active if Trms > 0.0.		
Qbunch	Real*8	nC	0.0
	bunch charge. Scaling is active if Qbunch $\neq$ 0.0.		
TRACK_ALL	Logical		TRUE
	if false, only the reference particle will be tracked.		

AUTO_PHASE	Logical		TRUE
	if true, the rf phases will be set relative to the phase with maximum energy gain.		
PHASE_SCAN	Logical		FALSE
	if true, the rf phases of the cavities will be scanned between 0 and 360 degree. Results are saved in the PScan file. The tracking between cavities will be done with the user-defined phases.		
H_max	Real*8	ns	0.001
	maximum time step for the Runge-Kutta integration.		
H_min	Real*8	ns	0.001
	minimum time step for the Runge-Kutta integration and min. time step for the space charge calculation. During the emission process from a cathode the time step is forced to H_min.		
Rmax	Real*8	m	1.0
	a particle will be discarded, if its radius (w.r.t. the z-axis of the coordinate system) exceeds Rmax.		
debunch	Real*8		10.0
	‘Debunched’ particles, i.e. particles with a distance to the bunch center exceeding debunch$\cdot\sigma z$ are passivated, i.e. their status will be set 0 or 1. Debunch is defined relative to the rms bunch length.		
ZSTART	Real*8	m	0.0
	minimal z postion for the generation of output, tracking may start at $z \neq$ ZSTART according to the definition of the initial particle distribution.		
ZSTOP	Real*8	m	0.3
	tracking will stop when the reference particle (the bunch center) passes ZSTOP.		
Zemit	Integer		100
	the interval ZSTOP-ZSTART is divided into Zemit subintervals. At the end of each subinterval output of line-typ is generated.		
Zphase	Integer		1
	the interval ZSTOP-ZSTART is divided into Zphase subintervals. At the end of each subinterval a complete particle distribution is saved. It is recommended to set Zemit = n·Zphase, $n \in N$.		
Lmagnetized	Logical		FALSE
	if true, solenoid fields are neglected in the calculation of the beam emittance.		
Lsub_rot	Logical		FALSE
	if true Lmagnetized will be set to true and the rotation of the bunch is subtracted based on the actual rotation of the bunch rather than by the canonical momentum.		
Lproject_emit	Logical		FALSE
	if true, the transverse particle positions of all particles will be projected into a common plane at the longitudinal bunch center position prior to the calculation of the emittance, spot size etc.		

RefS	Logical		FALSE
	if true, output files according to Table 3 and Table 4 are generated.		
EmitS	Logical		FALSE
	if true, output files according to Table 3 and Table 4 are generated.		
C_EmitS	Logical		FALSE
	if true, output files according to Table 3 and Table 4 are generated.		
PhaseS	Logical		FALSE
	if true, output files according to Table 3 and Table 4 are generated.		
TrackS	Logical		FALSE
	if true, output files according to Table 3 and Table 4 are generated.		
TcheckS	Logical		FALSE
	if true, output files according to Table 3 and Table 4 are generated.		
CathodeS	Logical		FALSE
	if true, output files according to Table 3 and Table 4 are generated.		

8.2.0 The namelist SCAN

In the namelist SCAN parameters for the scanning procedure are specified. See chapter 4.3.0.

Parameter	Specification	Unit	Default Value
LOOP	Logical		FALSE
	see chapter 4.3.0.		
NLoop	Integer		
	see chapter 4.3.0.		
LScan	Logical		FALSE
	if true a scan will be performed.		
Scan_para	Character*24		
	parameter to be scanned. Valid are all parameters that are written in italic letters in the namelist tables of chapter 8.0.0.		
S_min	Real*8		
	minimal set point of the scanning parameter.		
S_max	Real*8		
	maximal set point of the scanning parameter.		
L_min	Logical		FALSE
	if true, the minimum value of FOM(1) in the interval S_zmin to S_zmax will be saved. The value of FOM(2)-FOM(10) will be saved at the position of the minimum of FOM(1).		
L_max	Logical		FALSE
	if true, the maximum value of FOM(1) in the interval S_zmin to S_zmax will be saved. The value of FOM(2)-FOM(10) will be saved at the position of the maximum of FOM(1).		
S_zmin	Real*8	m	
	minimum position for the search of an optimum of FOM(1)		
S_zmax	Real*8	m	
	maximum position for the search of an optimum of FOM(1)		
S_dz	Integer		
	the interval S_zmin to S_zmax is divided into S_dz equal intervals. At the end of each interval emittances etc are calculated.		

FOM( )	Character*24 array		
	specifies the output to be found in the scan file. Valid keywords are: bunch **charge** **hor**izontal rms **emit**tance **hor**izontal rms **spot** size **hor**izontal rms beam **div**ergence **hor**izontal beam **offset** **ver**tical rms **emit**tance **ver**tical rms **spot** size **ver**tical rms beam **div**ergence **ver**tical beam **offset** **long**itudinal rms **emit**tance rms bunch **length** **mean** beam **energy** **rms** beam **energy** horizontal core emittance 95% **Cx_95** horizontal core emittance 90% **Cx_90** horizontal core emittance 80% **Cx_80** vertical core emittance 95% **Cy_95** vertical core emittance 90% **Cy_90** vertical core emittance 80% **Cy_80** longitudinal core emittance 95% **Cz_95** longitudinal core emittance 90% **Cz_90** longitudinal core emittance 80% **Cz_80** **phi end**		

8.3.0 The namelist CHARGE

In the namelist CHARGE parameters for the space charge calculation are specified. See chapter 4.2.0.

Parameter	Specification	Unit	Default Value
LOOP	Logical		FALSE
	see chapter 4.3.0.		
NLoop	Integer		
	see chapter 4.3.0.		
Lmirror	Logical		TRUE
	if true, mirror charges at the cathode are taken into account, see chapter 4.2.2.		
Linert	Logical		FALSE
	if true, magnetic fields in the average rest system of the bunch are taken into account, see chapter 4.2.3.		
LSPCH	Logical		FALSE
	if false, the space charge fields are not taken into account.		
Nrad	Integer		10
	number of grid cells in radial direction up to the bunch radius.		
Nlong_in	Integer		10
	maximum number of grid cells in longitudinal direction within the bunch length. During the emission process the number is reduced, according to the specification of the minimum cell length min_grid.		
Cell_var	Real*8		1.0
	variation of the cell height in radial direction. The innermost cell is cell_var times higher than the outermost cell.		
min_grid	Real*8	m	0.5D-6
	minimum grid length during emission. If the bunch is too short the number of cells is reduced accordingly.		
Max_scale	Real*8		0.1
	if one of the space charge scaling factors exceeds the limit 1± max_scale a new space charge calculation is initiated.		
Max_count	Integer		100
	if the space charge field has been scaled max_count times a new space charge calculation is initiated.		

8.4.0 The namelist CAVITY

The namelist CAVITY allows to include arbitrary rf fields. Standing wave structures and static electric fields are defined by means of tables, which may be generated by analytical calculations, measurements or numerical codes. The table has to contain the z-position (column1 in m) and the corresponding longitudinal on-axis electric field amplitude (column 2 in arb. units) in a free format.

In case of a static field set the frequency to 0 and the phase to 90°, so that the magnetic field component is zero.

The generation of a travelling wave structure is based on analytical expressions. The field has smooth edges therefore the actual field is somewhat longer than the effective length specified by the user.

Parameter	Specification	Unit	Default Value
LOOP	Logical		FALSE
	see chapter 4.3.0.		
NLoop	Integer		
	see chapter 4.3.0.		
LEfield	Logical		FALSE
	if false, all cavity fields are turned off.		
File_Efield( )	Character*80 array		
	user specified file name in case of a standing wave structure or a static field. To generate a travelling wave structure set File_Efield( ) to **TWS**		
C_noscale( )	Logical		FALSE
	if true, the cavity field will not be scaled, but the file values will be taken as field values in MV/m .		
Nue( )	Real*8 array	GHz	
	frequency of the rf field.		
K_wave( )	Real*8 array	m^{-1}	
	wave number of the field, only for β matched travelling wave structures.		
MaxE( )	Real*8 array	MV/m	
	maximum field amplitude of the rf field. The field is scaled to this value.		
Phi( )	Real*8 array	degree	
	phase of the rf field. Phi(0) defines a global phase shift of all cavities. See chapter 4.1.0.		
C_pos( )	Real*8 array	m	
	shifts the cavity position. C_pos is added to the position defined in File_Efield( ). In case of a travelling wave structure C_pos defines the entrance of the structure.		
C_length( )	Real*8 array	m	
	effective length of the cavity, only for travelling wave structures.		

8.5.0 The namelist SOLENOID

The namelist SOLENOID allows to include arbitrary solenoid fields by means of tables, which may be generated by analytical calculations, measurements or numerical codes. The table has to contain the z-position (column1 in m) and the corresponding longitudinal on-axis magnetic field amplitude (column 2 arb.units) in a free format.

Parameter	Specification	Unit	Default Value
LOOP	Logical		FALSE
	see chapter 4.3.0.		
NLoop	Integer		
	see chapter 4.3.0.		
LBfield	Logical		FALSE
	if false, all solenoid fields are turned off.		
File_Bfield( )	Character*80 array		
	user specified file name.		
S_noscale( )	Logical		FALSE
	if true, the solenoid field will not be scaled, but the file values will be taken as field values in T .		
MaxB( )	Real*8 array	T	0.0
	maximum field value of the solenoid field. The field is scaled to this value.		
S_xoff( )	Real*8 array	m	0.0
	horizontal offset of the solenoid.		
S_yoff( )	Real*8 array	m	0.0
	vertical offset of the solenoid.		
S_pos( )	Real*8 array	m	0.0
	shifts the solenoid position. S_pos is added to the position defined in File_Bfield( ).		

8.6.0 The namelist QUADRUPOLE

The namelist QUADRUPOLE allows to include quadrupole fields based on analytical expressions. The field has smooth edges therefore the actual field is somewhat longer than the effective length specified by the user. The tapering is user-defined by means of the parameter Q_bore, which might be set approximately to the radius of the quadrupole bore.

Parameter	Specification	Unit	Default Value
LOOP	Logical		FALSE
	see chapter 4.3.0.		
NLoop	Integer		
	see chapter 4.3.0.		
Lquad	Logical		FALSE
	if false, all quadrupole fields are turned off.		
Q_length( )	Real*8 array	m	
	effective length of the quadrupole.		
Q_type( )	Character*16 array		
	besides normal quadrupoles (w.o. type specification) **sk**ew quadrupoles, **do**ublets and **tr**iplets can be specified. Doublets have the same field amplitude with reversed sign in the two magnets. Triplets have an opposite sign in the outer magnets as compared to the inner. The field amplitude is the same for all magnets but the inner magnet is twice as long as the outer ones.		
Q_grad( )	Real*8 array	T/m	
	quadruple gradient. Refers to the first quadrupole in case of doublets and triplets.		
Q_K( )	Real*8 array	m^{-2}	
	focusing strength of the quadrupole. The gradient is set during the tracking of the reference particle in dependence of the beam energy. Refers to the first quadrupole in case of doublets and triplets.		
Q_bore( )	Real*8 array	m	
	taper parameter for the quadrupole field edge.		
Q_dist( )	Real*8 array	m	
	distance between magnets in case of doublets and triplets.		
Q_pos( )	Real*8 array	m	
	longitudinal quadrupole position. Refers to the center of the magnet(s) also in case of doublets and triplets.		
Q_xoff( )	Real*8 array	m	
	horizontal offset of the quadrupole, the doublet or the triplet.		
Q_yoff( )	Real*8 array	m	
	vertical offset of the quadrupole, the doublet or the triplet.		